# Thoughts on implementing ATLAS-style background subtraction

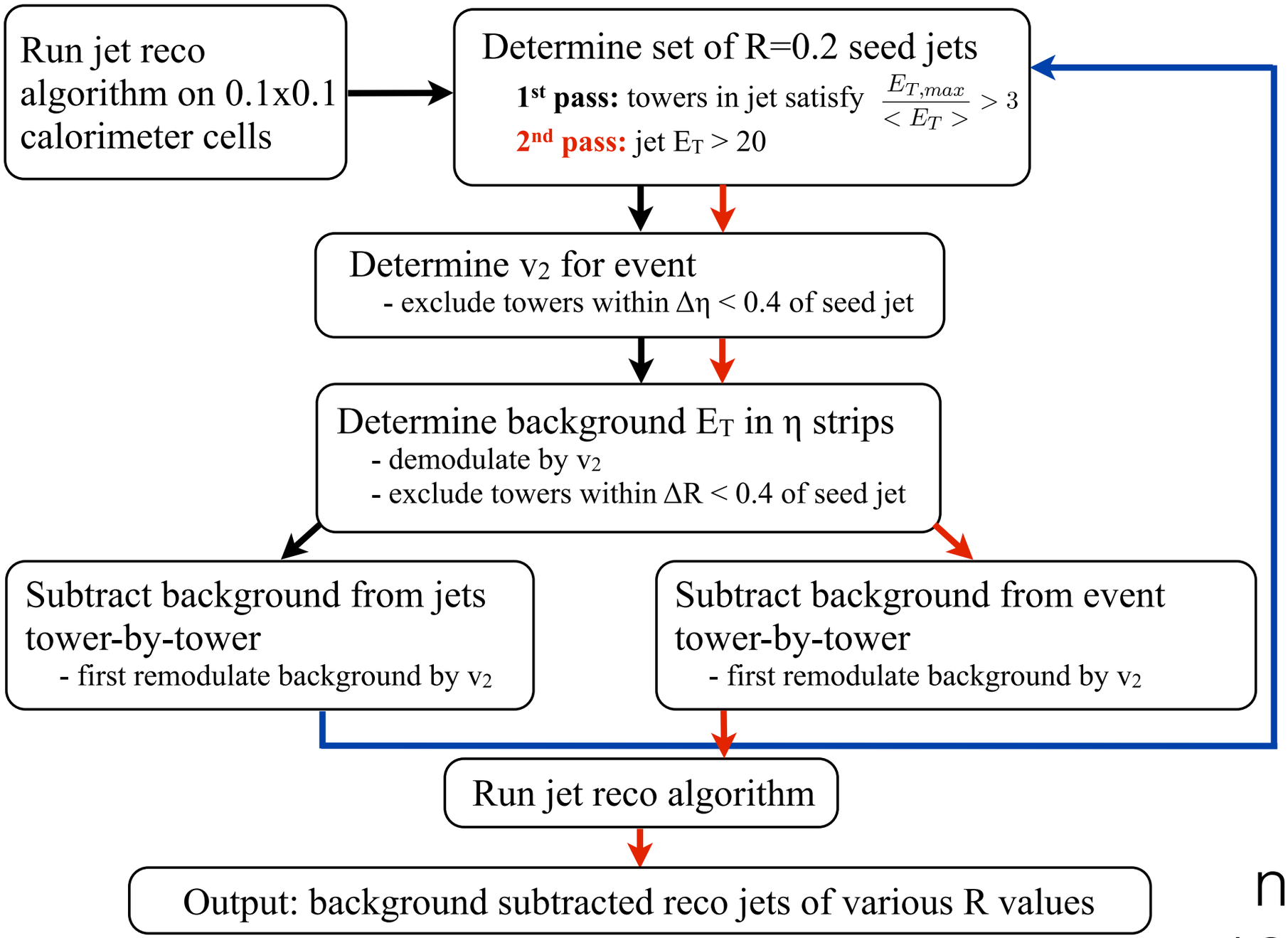


FIG. 1: Schematic illustration of the jet background subtraction method.

nucl-ex/1203.1353

# Some questions

- d$E_T$/dη and $v_2$ estimation in how many layers?
  - ➡ 1 layer: create 0.1x0.1-sized grid based on HCal geometry, partition energy in EMCal towers based on which HCal towers they overlap with
  - ➡ 2: deal with EMCal and IHCal+OHCal separately, makes sense since they have different geometries / different background levels
  - ➡ 3: each layer separately: seems like over-complication, but this is what's currently done in ATLAS…
- How much of existing jet reco modules / software to use?
  - ➡ example #1: create parallel copies of CEMC / HCALIN / HCALOUT tower containers, at different levels of subtraction, feed into JetReco module as normal
  - ➡ example #2: create purpose-built, lightweight containers and jet reco modules

# Possible software design

- Containers which live on the node tree:
  - ➡ BackgroundContainer: stores background $dE_T/d\eta$ vs. η, and global $v_2$ values
  - ➡ FullCaloTowerContainer: stores $d^2E_T/d\eta d\phi$ in calo towers at various levels of subtraction (or is this too similar to existing tower containers?)
  - ➡ HIJetContainer: stores HI jet (and constituent) information (or is this too similar to existing jet container?)
- Distinct Fun4All Modules:
  - ➡ **ConstructFullCaloTowers**: run only one, initiates the "raw" FullCaloTowerContainer from existing CEMC / HCALIN / HCALOUT tower information
  - ➡ **RunHIJetReco**: given a FullCaloTowerContainer, puts a set of reconstructed jets in an HIJetContainer on the node tree (potentially of various *R* sizes)
  - ➡ **DetermineBackground**: takes the raw FullCaloTowerContainer and a set of jets to be used for the seed exclusion (& seed definition), estimates the background and puts the resulting BackgroundContainer on the node tree
  - ➡ **GenerateFullCaloTowers**: given an existing FullCaloTowerContainer and the desired BackgroundContainer, puts a subtracted set of towers in a FullCaloTowerContainer on the node tree

# Possible HI jet reco flow

1. ConstructFullCaloTowers() → creates a FullCaloTowerContainer **raw_calo**
2. RunHIJetReco( FullCaloTowerContainer **raw_calo** , *R*=0.2 only ) → creates a HIJetContainer **first_jets**
3. DetermineBackground( FullCaloTowerContainer **raw_calo**, HIJetContainer **first_jets** ) → using *R*=0.2 *D*>3 jets as seeds, creates a background object BackgroundContainer **first_bkg**
4. GenerateFullCaloTowers( FullCaloTowerContainer **raw_calo**, BackgroundContainer **first_bkg** ) → creates an initial subtracted set of towers FullCaloTowerContainer **second_calo**
5. RunHIJetReco( FullCaloTowerContainer **second_calo** , *R*=0.2 only) → creates a HIJetContainer **second_jets**
6. DetermineBackground( FullCaloTowerContainer **raw_calo**, HIJetContainer **second_jets** ) → using *R*=0.2 pT > 20 GeV jets from 2nd pass as seeds, creates a background object BackgroundContainer **second_bkg**

   ➡ note: this goes back to the raw (initial) calo
7. GenerateFullCaloTowers( FullCaloTowerContainer **raw_calo**, BackgroundContainer **second_bkg** ) → creates the final set of subtracted towers FullCaloTowerContainer **final_calo**
8. RunHIJetReco( FullCaloTowerContainer **final_calo** , *R=0.2,0.3,0.4,0.5*) → creates a HIJetContainer **final_jets**